IN THE CLAIMS

1 – 57. (Cancelled)

58. (Previously Presented) A method for mapping a texture onto a surface of a computer generated object comprising the steps of:

approximating a true pixel color by performing a number of texturing operations, said texturing operations being determined by a geometric shape of a projection of a pixel on the texture; and

averaging results of said texturing operations.

59. (Previously Presented) A method as set forth in claim 58, wherein each of said texturing operations comprises:

accessing a mipmap at least one time; and

responding to multiple accesses being performed by interpolating results of the accesses.

60. (Previously Presented) A method as set forth in claim 59, wherein said number of texturing operations is a power of two.

61. (Previously Presented) A method as set forth in claim 60, wherein said number of texturing operations is less than or equal to a predetermined limit.

62. (Previously Presented) A method as set forth in claim 59, wherein the texture represents a reflected environment.

63. (Previously Presented) A method as set forth in claim 59, further comprising modifying a specularly reflected light intensity on the surface by combining said specularly reflected light intensity with a specular reflectance coefficient, said specular reflectance coefficient being retrieved from a specular reflectance coefficient map associated with the surface.

64. (Previously Presented) A method as set forth in claim 63, wherein combining said specularly reflected light intensity with a specular reflectance coefficient comprises multiplying said specularly reflected light intensity by the specular reflectance coefficient.

65. (Previously Presented) A method for modifying a specularly reflected light intensity on a surface of a computer generated object, comprising:

combining the specularly reflected light intensity with a specular reflectance coefficient, said specular reflectance coefficient being retrieved from a specular reflectance coefficient map associated with a texture map.

66. (Previously Presented) A method as set forth in claim 65, wherein combining the specularly reflected light intensity with the specular reflectance coefficient comprises multiplying the specularly reflected light intensity by the specular reflectance coefficient.

67. (Previously Presented) A texturing unit for mapping a texture to a surface of a computer generated object, which texture comprises a mipmap, which mipmap comprises a plurality of levels, each of which levels comprises at least one texel, the texturing unit comprising:

a control unit for receiving an input signal and determining a set of N footprint texel locations and at least one footprint level of detail from the input signal, which input signal includes information about a location and a shape of a projection of a pixel on the texture;

a Random Access Memory (RAM) coupled to the control unit for storing information representing the texture, receiving the set of N footprint texel locations and the footprint level of detail from the control unit, and determining N sets of texel values, where each set of texel values is associated with one footprint texel location, and where each set of texel values includes at least one texel value;

an interpolator coupled to the RAM, for accepting from the RAM the N sets of texel values and interpolating N interpolated values therefrom;

an averaging unit coupled to the interpolator for accepting from the interpolator the N interpolated values and determining an averaged value therefrom; and

an output port coupled to the averaging unit, for transmitting the averaged value to a device coupled to the output port.

68. (Previously Presented) The texturing unit of claim 67, further comprising:

a mipmap generation unit, coupled to the RAM, for accepting a changing video image, for generating a generated mipmap in real-time based on the changing video image, and for putting the generated mipmap into the RAM.

69. (Previously Presented) The texturing unit of claim 68, wherein the changing video image is an interlaced video image and the texturing unit further comprises:

a memory coupled to the mipmap generation unit for holding an interlaced half-frame of the interlaced video image.

70. (Previously Presented) The texturing unit of claim 68, wherein the mipmap generation unit calculates each level of the generated mipmap incrementally based on available information from the next level of higher detail.

71. (Previously Presented) An electronically-readable medium storing a program for permitting a computer to perform a method comprising:

approximating a true pixel color by performing a number of texturing operations, said texturing operations being determined by a geometric shape of a projection of a pixel on the texture, each of said texturing operations including accessing a mipmap at least one time in a marching direction corresponding to the geometric shape of the projection of the pixel on the texture; and

averaging results of said texturing operations.

72. (Previously Presented) An electronically-readable medium storing a program for permitting a computer to perform a method for modifying a specularly reflected light intensity on a surface of a computer generated object, the method comprising:

combining the specularly reflected light intensity with a specular reflectance coefficient, said specular reflectance coefficient being retrieved from a specular reflectance coefficient map associated with a texture map.

73. (Previously Presented) An electronically-readable medium storing a program for permitting a computer to perform a method for adding detail to a texture map comprising at least one texture element, the method comprising:

generating a detail map;

assigning a pointer into said detail map to at least one of the texture elements of the texture map to generate a pointer map, said pointer comprising two offsets including a first offset stored in a first offset map and a second offset stored in a second offset map;

interpolating detail color based on the generated detail map;

interpolating texture color based on the texture map; and

combining detail color with texture color to generate a pixel color.

<u>74. (New) A method as set forth in claim 58, wherein performing a number of texturing operations further comprises approximating the projection of the pixel on the texture by a footprint assembly.</u>

75 (New) A method as set forth in claim 74, wherein the footprint assembly comprises a plurality of texels.

76 (New) A method as set forth in claim 75, wherein each texel is a square mipmapped texel.

77. (New) A method as set forth in claim 74, wherein the geometric shape of the projection of the pixel on the texture is a parallelogram comprising a larger side and a smaller side, and wherein approximating the projection of the pixel further comprises generating coordinates for each texel based on a direction along the larger side of the parallelogram.